Final Report on ONR N00014-94-1-1137
Evolving Algebras:
A Novel Specification and Verification Methodology

Principal Investigator: Professor Yuri Gurevich
Electrical Engineering and Computer Science Department
The University of Michigan
Ann Arbor, MI 48109-2122
gurevich@umich.edu

Start/End Dates: 9/1/94 — 12/31/97

The Evolving Algebra Methodology originated at Michigan and, from the beginning, was supported by ONR. Recently the term "Evolving Algebra" has been replaced with the term "Abstract State Machines" abbreviated to ASM. During the period in question, the ASM theory has been advanced and many ASM applications have been successfully realized. The ASM methodology spread to many countries including many European countries, e.g. France, Germany, Switzerland. Annual ASM workshops have been established. The latest ASM workshop took place in June 1997 in France; see http://www.tik.ee.ethz.ch/ kutter/ASMworkshop/1997/. Two ASM workshop will take place in 1998: one in France and one in Germany. General information about the current state of affairs in the ASM field can be found at http://www.eecs.umich.edu/gasm/. An ASM interpreter has been improved here at Michigan; additional ASM tools have been; in particular see http://www.icsi.berkeley.edu/ maffy/gem. In the rest of the report, we provide an annotated list of the articles of the principal investigator and his students written, at least partially, during the period in question.

Articles

1. Yuri Gurevich
   Evolving Algebra 1993: Lipari Guide
   in "Specification and Validation Methods"
   Ed. E. Boerrger, Oxford University Press, 1995, 9–36.

   Computation models and specification methods seem to be worlds apart. The evolving algebra project started as an attempt to bridge the gap by improving on Turing's thesis. We sought more versatile machines which would be able to simulate arbitrary algorithms, *on their natural abstraction levels*, in a direct and essentially coding-free way. The evolving algebra thesis asserts that evolving algebras are such versatile machines. The guide provides the definition of sequential and – for the first time – concurrent and distributed evolving algebras. In the same volume, Egon Boerrger gives an annotated bibliography of evolving algebra applications.

2. Yuri Gurevich and Raghu Mani
Group Membership Protocol: Specification and Verification
in "Specification and Validation Methods"
Ed. E. Boerger, Oxford University Press, 1995, 295–328.

The interesting and useful protocol of Flavio Christian involves timing constraints, and its correctness is not obvious. We formally verify the protocol (and notice that tbe assumptions about the environment may be somewhat weakened).

3. James K. Huggins
"Kermit: Specification and Verification"
in "Specification and Validation Methods"
ed. E. Börger, Oxford University Press, 1995, 247–293.

This is a part of Jim Huggins's PhD thesis. He gives a mathematical model of the well-known Kermit communication protocol and proves the correctness of the protocol.

4. Egon Boerger, Dean Rosenzweig and Yuri Gurevich
Lamport's Bakery Algorithm
in "Specification and Validation Methods"
Ed. E. Boerger, Oxford University Press, 1995, 231–243.

An evolving algebra A1 is constructed to reflect faithfully the algorithm. Then a more abstract ealgebra A2 is constructed. We check that A2 is safe and fair, and that A1 correctly implements A2.

5. Charles Wallace
The semantics of the C++ programming language
In Specification and Validation Methods
E. Boerger, editor, pages 131-164. Oxford University Press, 1995.

The mathematical model of the C programming language by Gurevich and Huggins is extended to C++.

6. Yuri Gurevich, Neil Immerman and Saharon Shelah
McColm's Conjecture
Symposium on Logic in Computer Science, IEEE Computer Society Press, 1994, 10–19.

Gregory McColm conjectured that positive elementary inductions are bounded in a class K of finite structures if every FO(LFP) formula is equivalent to a first-order formula in K. Here FO(LFP) is the extension of first-order logic with the least fixed point operator. We disprove the conjecture. Our main results are two model-theoretic constructions — one deterministic and one probabilistic — each of which refutes McColm's conjecture.

7. Erich Graedel and Yuri Gurevich
Metafinite Model Theory
in D. Leivant (Ed.), Logic and Computational Complexity, Selected Papers,
Lecture Notes in Computer Science Nr. 960, Springer 1995, 313–366.
To appear in a special issue of Information and Computation.

In [60], we promoted the use of finite structures to model databases. In a Hegelian twist, we coclude here that finite structures may be inadequate to model databases

and other dynamic systems, and we suggest to extend the approach and methods of finite model theory beyond finite structures. One particular motivation of the generalization is that objects consisting of both finite structures and numbers from infinite domains appear again and again in computer science applications. We study definability issues and their relation to complexity on *metafinite structures* which typically consist of (i) a primary part, which is a finite structure, (ii) a secondary part, which is a (usually infinite) structure that can be viewed as a structured domain of numerical objects, and (iii) a set of "weight" functions from the first part into the second. We discuss model-theoretic properties of metafinite structures, present results on descriptive complexity, and sketch some potential applications.

8. Andreas Blass and Yuri Gurevich
   Evolving Algebras and Linear Time Hierarchy
   in "IFIP 1994 World Computer Congress, Volume I:
   Technology and Foundations"
   eds. B. Pehrson and I. Simon, North-Holland, Amsterdam, 383–390.

   A precursor of [16]

9. Yuri Gurevich and James K. Huggins
   Evolving Algebras and Partial Evaluation
   in "IFIP 1994 World Computing Congress, Volume 1: Technology and Foundations"
   eds. B. Pehrson and I. Simon, Elsevier, Amsterdam, 587–592.

   We describe an automated partial evaluator for sequential evolving algebras implemented at the University of Michigan.

10. Yuri Gurevich
    Evolving Algebras
    in "IFIP 1994 World Computer Congress, Volume I: Technology and Foundations"
    eds. B. Pehrson and I. Simon, Elsevier, Amsterdam, 423–427.

    The opening talk at the first EA workshop. Sections: Introduction, The EA Thesis, Remarks, Future Work.

11. Yuri Gurevich and Saharon Shelah
    On Rigid Structures
    Journal of Symbolic Logic, vol. 61, no. 2, June 1996, 549–562.

    The main result is a probabilistic construction of finite rigid graphs. Using the construction, we exhibit a finitely axiomatizable class of finite rigid structures such that no $L^{\omega}_{\infty,\omega}$ sentence with counting quantifiers defines a linear order in every structure of the class.

12. Yuri Gurevich
    The Value, if any, of Decidability
    Bulletin of European Assoc. for Theor. Computer Sci., Feb. 1995, 129–135.

    A decidable problem can be as hard as an undecidable one for practical purposes. So what is the value of a mere decidablity result? That is the topic discussed in the paper.

13. Thomas Eiter, Georg Gottlob and Yuri Gurevich
Normal Forms for Second-Order Logic over Finite Structures,
and Classification of NP Optimization Problems
Annals of Pure and Applied Logic, 78 (1996), 111–125.

We prove a new normal form for second-order formulas on finite structures and simplify the Kolaitis-Thakur hierarchy of NP optimization problems.

14. Yuri Gurevich and James K. Huggins
The Railroad Crossing Problem:
An Experiment with Instantaneous Actions and Immediate Reactions,
in "Computer Science Logics, Selected papers from CSL'95"
ed. H. K. Buening, Springer Lecture Notes in Computer Science
Vol. 1092, 1996, 266–290.

We give an evolving algebra solution for the well-known railroad crossing problem and use the occasion to experiment with agents that perform instantaneous actions in continuous time and in particular with agents that fire at the moment they are enabled.

15. Yuri Gurevich and James K. Huggins
Equivalence is in the Eye of the Beholder
Theoretical Computer Science, to appear.

In a recent provocative paper, Lamport points out "the insubstantiality of processes" by proving the equivalence of two different decompositions of the same intuitive algorithm by means of temporal formulas. We point out that the equivalence of algorithms is itself in the eye of the beholder. We discuss a number of related issues and, in particular, demonstrate that algorithms can be proved equivalent directly.

16. Andreas Blass and Yuri Gurevich
The Linear Time Hierarchy Theorems for RAMs and Abstract State Machines
Springer J. of Universal Computer Science
Vol. 3, No. 4, April 28, 1997, 247–278.

Contrary to polynomial time, linear time greatly depends on the computation model. In 1992, Neil Jones designed a number of computation models where the linear-speed-up theorem fails and linear-time computable functions form a hierarchy. The linear time of those models is too restrictive. We prove linear-time hierarchy theorems for random access machines and Gurevich abstract state machines (formerly evolving algebras). The latter generalization is harder and more important because of the greater flexibility of the ASM model. One long-term goal of this line or research is to prove lower bounds for natural linear time problems.

17. Yuri Gurevich and Marc Spielmann
Recursive Abstract State Machines
Springer J. of Universal Computer Science
Vol. 3, No. 4, April 28,1997, 233–246.

The Gurevich abstract state machine (ASM) thesis, supported by numerous applications, asserts that ASMs express algorithms on their natural abstraction levels directly and essentially coding-free. The only objection raised to date has been that ASMs

are iterative in their nature, whereas many algorithms are naturally recursive. There seems to be an inherent contradiction between
(i) the ASM idea of explicit and comprehensive states, and
(ii) higher level recursion with its hiding of the stack.
But consider recursion more closely. When an algorithm A calls an algorithm B, a clone of B is created and this clone becomes a slave of A. This raises the idea of treating recursion as an implicitly distributed computation. Slave agents come and go, and the master/slave hierarchy serves as the stack.

Building upon this idea, we suggest a definition of recursive ASMs. The implicit use of distributed computing has an important side benefit: it leads naturally to concurrent recursion. In addition, we reduce recursive ASMs to distributed ASMs. If desired, one can view recursive notation as mere abbreviation.

18. Andreas Blass, Yuri Gurevich and Saharon Shelah
Choiceless Polynomial Time
Tech. Report, EECS Dept, U. of Michigan, May 1997.

Turing machines define polynomial time (PTime) on strings but cannot deal with structures like graphs directly, and there is no known, easily computable string encoding of isomorphism classes of structures. Is there a computation model whose machines do not distinguish between isomorphic structures and compute exactly PTime properties? This question can be recast as follows: Does there exists a logic that captures polynomial time (without presuming the presence of a linear order)? Earlier, one of us conjectured the negative answer; see [74]. The problem motivated a quest for stronger and stronger PTime logics. All these logics avoid arbitrary choice. Here we attempt to capture the choiceless fragment of PTime. Our computation model is a version of abstract state machines (formerly called evolving algebras). The idea is to replace arbitrary choice with parallel execution. The resulting logic is more expressive than other PTime logics in the literature. A more difficult theorem shows that the logic does not capture all PTime.

19. Scott Dexter, Patrick Doyle and Yuri Gurevich
Gurevich Abstract State Machines and
Schoenhage Storage Modification Machines
Springer J. of Universal Computer Science
Vol. 3, No. 4, April 28, 1997, 279–303.

We show that, in a strong sense, Schoenhage's storage modification machines are equivalent to the unary fragment of basic abstract state machines (without external functions). The unary restriction can be removed if the storage modification machines are equipped with a pairing function in an appropriate way.

20. Charles Wallace, Yuri Gurevich and Nandit Soparkar
A Formal Approach to Recovery in Transaction-Oriented Database Systems
Springer J. of Universal Computer Science
Vol. 3, No. 4, April 28, 1997, 320–340.

Previous version: Same authors
Formalizing Recovery in Transaction-Oriented Database Systems
Proc. of COMAD'95 (Seventh International Conference on Management of Data)

eds. S. Chaudhuri, A. Deshpande, and R. Krishnamurthy
New Delhi, India, Tata McGraw-Hill, 1995, 166–185.

Failure resilience is an essential requirement for transaction-oriented database systems, yet there has been little effort to specify and verify techniques for failure recovery formally. The desire to improve performance has resulted in algorithms of considerable sophistication, understood by few and prone to errors. In this paper, we show how the formal methodology of Gurevich Abstract State Machines can elucidate recovery and provide formal rigor to the design of a recovery algorithm. In a series of refinements, we model recovery at several levels of abstraction, verifying the correctness of each model. This initial work indicates that our approach can be applied to more advanced recovery mechanisms.

21. Yuri Gurevich
Platonism, Constructivism, and Computer Proofs vs. Proofs by Hand
Bull. of Euro. Assoc. of Theor. Computer Science, Oct. 1995, 145–166.

For many years, constructivists criticized classical mathematics. For once, the favor is returned.

22. Natasha Alechina and Yuri Gurevich
Syntax vs. Semantics on Finite Structures
to appear in Springer Lecture Notes in Computer Science
tech. report CSR-96-14, School of Computer Science, U. of Birmingham
available by ftp from http://www.cs.bham.ac.uk/ nxa/draft.ps.gz

Logic preservation theorems often have the form of a syntax/semantics correspondence. For example, the Los-Tarski theorem asserts that a first-order sentence is preserved by extensions if and only if it is equivalent to an existential sentence. Many of these correspondences break when one restricts attention to finite models. In such a case, one may attempt to find a new semantical characterization of the old syntactical property or a new syntactical characterization of the old semantical property. The goal of this paper is to provoke such a study.

23. Anatoli Degtiarev, Yuri Gurevich and Andrei Voronkov
"Herbrand's Theorem and Equational Reasoning: Problems and Solutions"
Bulletin of Euro. Assoc. for Theor. Computer Science
Vol. 60, Oct 1996, 78–95.

The article (written in a popular form) explains that a number of different algorithmic problems related to Herbrand's theorem happen to be equivalent. Among these problems are the intuitinistic provability problem for the existential fragment of first-order logic with equality, the intuitionistic provability problem for the prenex fragment of first-order with equality, and the simultaneous rigid E-unification problem (SREU). The article explains an undecidability proof of SREU and decidability proofs for special cases. It contains an extensive bibliography on SREU.

24. Yuri Gurevich and Margus Veanes
"Some Undecidable Problems Related to the Herbrand Theorem"
Tech. Report no. 138, March 1997
Computing Science Dept, Uppsala University, Sweden.

25. A. Degtyarev, Y. Gurevich, P. Narendran, M. Veanes and A. Voronkov
"The Decidability of Simultaneous Rigid E-Unification with One Variable"
Tech. Rep. 139, March 1997, Computing Sci. Dept, Uppsala Uni.,Sweden
submitted to RTA'98, 9th Conf. on Rewriting Techniques and Applications
Tsukuba, Japan, March 30 — April 1, 1998
A fuller version submitted to Theoretical Computer Science

    The title problem is proved decidable and in fact EXPTime complete. Furthermore, the problem becomes PTime complete if the number of equations is bounded by any (positive) constant. Notice that simultaneous rigid E-unification with two variable and only three equations is undecidable [126]. It follows that the A*EA* fragment of intuitionistic logic is decidable which contrasts with the undecidability of the EE fragment proved recently by Veanes.

    The full version, in addition to full proofs, includes also a decidability proof for the case for the case of simultaneous rigid E-unification when each rigid equation either contains (at most) one variable or else has a ground left-hand side and the right-hand side of the form x=y where x and y are variables.

26. Yuri Gurevich and Andrei Voronkov
"Monadic Simultaneous Rigid E-unification and Related Problems"
ICALP'97, International Colloquium on Automata, Languages and Programming, 1997.

    We study the monadic case of a decision problem known as simultaneous rigid E-unification. We show its equivalence to an extension of word equations. We prove decidability and complexity results for special cases of this problem.

27. Yuri Gurevich, "May 1997 Draft of the ASM Guide", Tech Report CSE-TR-336-97, EECS Dept, University of Michigan, 1997

    [This tech report appears here as an exception because it is used by the ASM community and it is not going to be published.]

28. Yuri Gurevich and Alex Rabinovich, "Definability and Undefinability with Real Order at the Background", submitted

    Let $\phi(X, Y)$ and $\psi(Y)$ range over formulas in the monadic second-order language of order. Let I be the set of integers and F be the family of subsets J of I such that $\phi(I, J)$ holds over the real line. The question arises whether, for every ϕ, the same family F can be defined by means of an appropriate $\psi(Y)$ interpreted over the integer order. We answer the question positively. Furthermore, the answer remains positive for every closed subset X of reals, but may be negative for some open subsets.

29. Yuri Gurevich, "From Invariants to Canonization", The Bull. of Euro. Assoc. for Theor. Computer Sci., no. 63, October 1997.

    We show that every polynomial-time full-invariant algorithm for graphs gives rise to a polynomial-time canonization algorithm for graphs.

30. Andreas Blass, Yuri Gurevich, Vladik Kreinovich and Luc Longpré, "A Variation on the Zero-One Law", submitted.

31. Erich Graedel, Yuri Gurevich and Colin Hirsch, "The Complexity of Query Reliability", submitted.

32. Thomas Eiter, Georg Gottlob and Yuri Gurevich, "Existential Second-Order Logic over Strings", submitted

33. Charles Wallace, "The Semantics of the Java Programmaing Language", Tech report CSE-TR-355-97, EECS Dept, University of Michigan.

    A mathematical model of the Java programmaing language is given.